*Appears in Proceedings of the 39th Annual International Symposium on Computer Architecture, Portland, OR, June 2012**

Scale-Out Processors

Pejman Lotfi-Kamran‡, Boris Grot‡, Michael Ferdman†,‡, Stavros Volos‡, Onur Kocberber‡,
Javier Picorel‡, Almutaz Adileh‡, Djordje Jevdjic‡, Sachin Idgunji*, Emre Ozer*, and Babak Falsafi‡
‡*EcoCloud, EPFL* †*CALCM, Carnegie Mellon* **ARM*

Abstract

Scale-out datacenters mandate high per-server throughput to get the maximum benefit from the large TCO investment. Emerging applications (e.g., data serving and web search) that run in these datacenters operate on vast datasets that are not accommodated by on-die caches of existing server chips. Large caches reduce the die area available for cores and lower performance through long access latency when instructions are fetched. Performance on scale-out workloads is maximized through a modestly-sized last-level cache that captures the instruction footprint at the lowest possible access latency.

In this work, we introduce a methodology for designing scalable and efficient scale-out server processors. Based on a metric of performance-density, we facilitate the design of optimal multi-core configurations, called pods. Each pod is a complete server that tightly couples a number of cores to a small last-level cache using a fast interconnect. Replicating the pod to fill the die area yields processors which have optimal performance density, leading to maximum per-chip throughput. Moreover, as each pod is a stand-alone server, scale-out processors avoid the expense of global (i.e., inter-pod) interconnect and coherence. These features synergistically maximize throughput, lower design complexity, and improve technology scalability. In 20nm technology, scale-out chips improve throughput by 5x-6.5x over conventional and by 1.6x-1.9x over emerging tiled organizations.

1. Introduction

Cloud computing has emerged as the foundation for scalable online services. Cloud operators, such as Google, Microsoft, and Facebook, rely on networks of datacenters to deliver search, social connectivity, and a growing number of other offerings. The *scale-out* software architecture at the core of the online service model effectively accommodates dataset and demand growth by simply adding more servers to the cloud, as servers handle independent requests that do not share any state. With typical scale-out applications distributed across thousands of servers inside a datacenter, performance characteristics of each server dictate the datacenter's throughput. In TCO-conscious datacenters, performance per TCO dollar is maximized by increasing the throughput of each server processor, which enables better memory utilization and affords higher per-server performance without a commensurate increase in cost [27].

Today's volume servers are designed with processors that are essentially general-purpose. These *conventional* processors combine a handful of aggressively speculative and high clock-frequency cores supplemented by a large shared on-chip cache. As manufacturing technology provides higher transistor density, conventional processors use the additional transistors to scale up the core count, cache capacity, and the coherence and interconnect layer.

Recently, *tiled* processors have emerged as competition to volume processors in the scale-out server space [26]. Recognizing the importance of per-server throughput, these processors use a large number of relatively simple cores, each with a slice of the shared LLC, interconnected via a packet-based mesh interconnect. Lower-complexity cores are more efficient than those in conventional designs [19]. Additionally, the many-core architecture improves throughput compared to conventional chips on memory- and I/O-bound scale-out workloads. Despite the differences in the chip-level organization, the technology scaling trends of tiled processors are similar to conventional designs; each technology generation affords more tiles, which increases the core count, cache capacity, and interconnect resources.

We observe that, in the context of processors for scale-out applications, both architectures make sub-optimal use of the die area. As recent research examining scale-out [7] and traditional server workloads [11] has demonstrated, large caches, such as those found both in conventional and tiled designs, are inefficient due to limited reuse at the LLC resulting from vast data footprints of these applications. In fact, large LLC configurations have been shown to be detrimental to performance, as they increase the fetch latency of performance-critical instructions whose footprint exceeds the capacity of first-level caches. Moreover, recent work has identified significant over-provisioning in conventional server chips' core capabilities, on-die interconnect, and memory bandwidth [7].

Our work confirms these results and shows that maximizing throughput necessitates a careful choice in the size of the cache. Smaller caches that can capture the dynamic instruction footprint of scale-out workloads afford more die area for the cores without penalizing per-core performance. Moreover, we demonstrate that while the simpler cores found in tiled designs are more effective than conventional server cores for scale-out workloads, the latency incurred by the on-chip interconnect in tiled organizations lowers performance and limits the benefits of integration, as additional tiles result in more network hops and longer delays.

In this work, we seek to develop a technology-scalable server chip architecture for scale-out workloads that makes optimal use of the die real-estate. We use *performance density (PD)*, defined as throughput per unit area, to quantify how effectively an architecture uses the silicon real-estate. We develop a design methodology to derive a performance-density optimal processor building block called a *pod*, which tightly couples a number of cores to a small LLC via a fast interconnect. As technology scales to allow more on-chip cores, our methodology calls for keeping the design of the pod unchanged, replicating the pod to use up the available die area and power budget. A key aspect of the proposed architecture is that pods are stand-alone servers, with no inter-pod connectivity or coherence. The pod methodology enables processors to scale seamlessly with technology, side-stepping the challenges of scaling both software and hardware to large core counts, while at the same time guaranteeing maximum throughput and optimally-efficient use of the on-chip real-estate.

We use analytic models and cycle-accurate full-system simulation of a diverse suite of representative scale-out workloads to demonstrate that:

- The core and cache area budget of conventional server processors is misallocated, resulting in a performance density gap of 3.4x to 6.5x against an optimally-efficient processor.
- The distributed cache architecture in tiled designs increases access latencies and lowers performance, as manifested in a performance density gap of 1.5x to 1.9x versus an optimally-efficient processor.
- Performance density can be used to derive an optimally-efficient pod that uses a small (i.e., 2-4MB) last-level cache and benefits from a high core-to-cache area ratio and simple crossbar interconnect.
- Replicating pods to fill the die results in an optimally-efficient processor which maximizes throughput and provides scalability across technology generations. For example, in the 20nm technology, scale-out processors improve performance density by 1.6x-6.5x over alternative organizations.

2. Motivation

We examine a representative set of scale-out applications in order to understand the demands they place on server architectures. In particular, we seek to establish the range of last-level cache sizes appropriate for these workloads and their sensitivity to contention in configurations where many cores share the LLC.

Our workloads are taken from CloudSuite [3] and represent the dominant applications in scale-out datacenters. Prior work [7] has shown that these applications have functionally similar characteristics, namely (a) they operate on huge data sets that are split across a large number of nodes into memory-resident shards; (b) the nodes service a large number of completely independent requests that do not share state; and (c) the inter-node connectivity is used only for high-level task management and coordination.

Sensitivity to LLC size. We first analyze the cache requirements of scale-out applications by sweeping the size of the last-level cache (LLC) from 1 to 32MB. We present the results for a quad-core CMP, but note that the general trends are independent of the core count. Details of the methodology can be found in Section 5.4.

Figure 1(a) plots the performance of individual applications normalized to a design with a 1MB LLC. For most of the workloads, we observe that LLC capacities of 2-8MB are sufficient to capture the instruction footprint and secondary working set. Beyond this range, larger cache configurations provide limited benefit because the enormous data working sets of the applications exhibit little reuse in the LLC. Two of the workloads (MapReduce-C and SAT Solver) exhibit a different behavior, as larger caches do help in capturing the secondary working set. However, even for these workloads, a 16-fold increase in cache size from 1 to 16MB translates into a performance gain of just 12-24%. Cache capacity beyond 16MB is strictly detrimental to performance, as the reduction in miss rate is offset by the increased access latency. These results corroborate prior characterizations of scale-out and traditional server workloads executing on chip multiprocessors [7, 11].

Sensitivity to core count. We analyze the sensitivity of scale-out applications to the number of threads and the sharing degree. We fix the LLC size at 4MB and examine the performance as the number of cores varies from 1 to 256. Figure 1 plots per-core performance (b) and throughput per chip (c) averaged across workloads and normalized to a single-core baseline. The two lines in the figures correspond to an ideal organization with a fixed-latency interconnect between each core and the LLC (solid grey line), and a realistic mesh-based interconnect where the physical distance between cores and cache banks affects the LLC access latency (dashed black line).

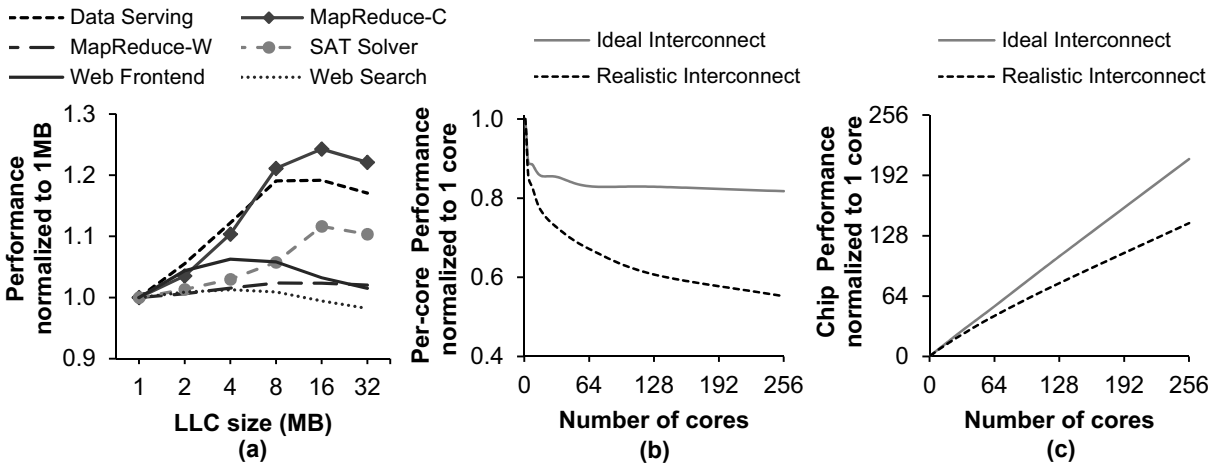


Figure 1. Performance of 4-core workloads varying the LLC size (a), per-core performance with a 4MB LLC varying the number of cores (b), and chip-level performance with a 4MB LLC varying the number of cores (c).

In the case of an ideal interconnect, Figure 1(b) shows that the degradation in per-core performance associated with having many cores share the LLC is small (e.g., 15% for a 128x increase in core count from 2 to 256 cores). As a result, Figure 1(c) demonstrates that aggregate performance can be improved by a factor of 210 by sharing a 4MB LLC among 256 cores.

In the case of a design subject to physical constraints in which the distance to the LLC grows with core count, the negative slope of the performance curve in Figure 1(b) is much steeper. The distance to the LLC has a direct effect on performance due to a combination of primary working set sizes greatly exceeding the L1 capacity and the memory-intensive nature of scale-out applications, which makes these applications particularly sensitive to the average memory access time. As a result, Figure 1(c) shows that a design based on a realistic interconnect reduces performance by 32% when compared to an ideal network at 256 cores, demonstrating how distance effects threaten the ability of server processors to reach their throughput potential.

Overall, scale-out applications show limited benefit from LLC capacities beyond 8MB. Furthermore, a moderately-sized cache can be effectively shared among a large number of cores. However, maximizing the performance in a system with a heavily-shared LLC requires mitigating interconnect delays.

3. Performance Density

Scale-out applications are inherently parallel and, consequently, best served by substrates that provide a large number of cores to achieve high per-server throughput. However, higher core density leaves less silicon real-estate for on-die caches, and at the same time increases the physical core-to-cache distance and interconnect delays. The cache should be large enough to capture the dynamic instruction footprint and shared OS data, yet small enough to provide fast access, which is particularly important for instruction fetches that lie on the critical path of execution. The physical distance between the cores and cache must also be short to minimize the delay due to the interconnect.

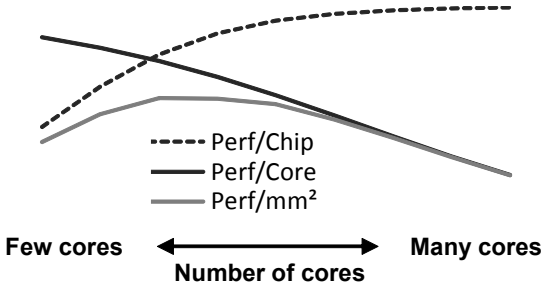
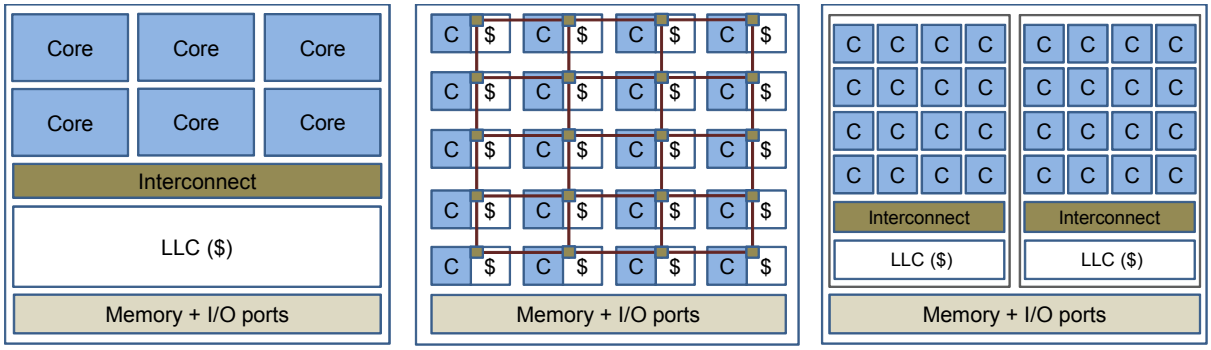


Figure 2. Performance per core, performance per chip, and performance density for a hypothetical workload.

To capture these conflicting requirements in a single metric for assessing processor efficiency, we propose *performance density (PD)*, defined as performance per mm^2. Given a core microarchitecture, PD provides a simple means of comparing a range of designs that differ in core count, LLC size, and interconnect parameters.

Figure 2 provides the intuition behind the performance density metric using a hypothetical workload whose behavior is representative of scale-out applications. The x-axis plots the number of cores for a fixed-size cache. The number of cores increases to the right of the graph, resulting in a higher core-to-cache ratio. The black solid line plots per-core performance, which diminishes as the number of cores grows due to the combination of distance and sharing at the LLC. The dashed line shows the aggregate throughput, which scales with the additional core resources, but the growth is sub-linear in core count due to the eroding per-core throughput. Finally, the gray line plots performance



(a) Conventional (b) Tiled (c) Scale-Out

Figure 3. Comparison of Conventional, Tiled, and Scale-Out architectures. The Scale-Out design features 2 pods.

density, whose peak represents an optimal configuration that maximizes performance per unit area by balancing core count, LLC capacity, sharing, and distance factors.

4. Scale-Out Processors

Today's server chips, such as the conventional processors and the emerging tiled designs, scale performance through the addition of cores, cache capacity, interconnect and coherence resources, and miscellaneous glue logic. This scaling strategy is a characteristic of the *scale-up* model. We find that the model of increasing processor complexity is counter-productive for scale-out workloads because additional resources do not yield a commensurate improvement in chip-level performance.

To overcome the limitations of the conventional design methodology, we develop a technology-scalable approach for maximizing the performance of server processors targeting scale-out workloads. Our approach uses performance density as an optimization metric and builds on a simple observation that, given a configuration that is PD-optimal, the most profitable way of scaling aggregate performance is to grow the number of PD-optimal units on chip. This strategy maintains the optimal performance density while increasing the aggregate throughput. In contrast, an approach that expands a PD-optimal configuration through additional core and cache resources lowers performance density and leads to a chip organization whose peak throughput is sub-optimal for a given area budget.

4.1. Overview

The notion at the heart of a scale-out processor is the *pod*, a PD-optimal organization of core, cache, and interconnect resources. A pod is a complete server that runs its own copy of the operating system. Depending on the characteristics of the underlying process technology and component microarchitecture, a single pod may require only a fraction of the available die area, power, and bandwidth budget. To fully leverage the benefits of integration, multiple pods can be placed on a die. In effect, a pod acts as the tiling unit in a scale-out processor.

Adding more pods does not affect the optimality of each individual pod, allowing performance to scale linearly with the pod count. Because each pod is a complete server-on-a-die, direct inter-pod connectivity is not required. Thus, perfect performance scalability comes at negligible integration expense that side-steps the challenge of scaling up the global interconnect and coherence infrastructure. The lack of inter-dependence among pods is a feature that fundamentally sets scale-out processors apart from existing chip organizations and enables optimality-preserving scaling across technology generations.

Figure 3 captures the spirit of our approach and highlights the differences from existing designs by comparing the chip-level organization of conventional, tiled, and scale-out designs. In the rest of this section, we explain the features of a PD-optimal pod, describe the implications of a pod-based design at the chip level, and briefly compare the performance density of scale-out processors to existing server chip organizations.

4.2. Pod Organization

We first examine a pod's cache requirements. Scale-out applications have large instruction working sets, in the range of one to several megabytes [7], which are not well accommodated by private caches [9]. A shared LLC is thus a better choice as it can capture the working set of application and OS instructions, along with OS and thread-private data, without the performance and area expense of per-core private cache hierarchies. However, once these elements are captured, larger LLC configurations do not benefit scale-out applications whose data sets greatly exceed capacities of practical on-die caches.

Because much of the useful capacity of the shared LLC comes from the common instruction and OS working set,

the cache is naturally amenable to high degrees of sharing, a trend shown in Figure 1. However, the high incidence of misses at the L1-I mandates an LLC organization with low access latency and a fast core-to-cache interconnect. Thus, performance density of a pod is maximized by balancing throughput gains arising from having many cores share an LLC against the reduction in per-core performance stemming from longer cache access latencies.

The core microarchitecture, cache parameters, and interconnect characteristics all play a role in determining a PD-optimal organization by influencing factors that include the cache bank access latency, core-to-cache distance, wire delays, and the pressure placed by each core on the cache. Across the spectrum of scale-out workloads, modest cache capacities in the range of 2-4MB are sufficient. Meanwhile, the number of cores required to maximize performance density for the range of parameters considered in our studies varies from 16 (for out-of-order cores) and up to 32 (for in-order cores).

4.3. Chip-Level Considerations

A Scale-Out chip is a simple composition of one or more pods and a set of memory and I/O interfaces. Multi-pod designs reduce the number of chips for a given throughput target or power budget. Having fewer chips on a motherboard is a plus in high-density datacenter racks, where motherboard space is limited and reducing the number of sockets per board may be an effective cost-reduction strategy.

Integrating multiple pods on a die improves efficiency by sharing the on-die DRAM and I/O ports, increasing bandwidth utilization and reducing pin requirement. Scale-out chips that share pins among pods necessitate a global interconnect layer to connect the individual pods to the memory and I/O ports. Fortunately, such a layer can be kept trivial due to the limited connectivity that it must provide, because pod-to-pod communication is not needed and the number of external interfaces is low.

The number of pods on a die is dictated by physical constraints, such as area, power, and pin bandwidth. The lack of inter-dependence among pods is a valuable feature that eliminates the need for pod-to-pod communication infrastructure and chip-wide coherence support. The absence of these mechanisms reduces the design complexity of scale-out processors and boosts their scalability.

4.4. Comparison to Existing Server Processors

We compare a scale-out processor to conventional and tiled designs in Figure 4. We show the total core area, cache (LLC) area, and average performance density across the

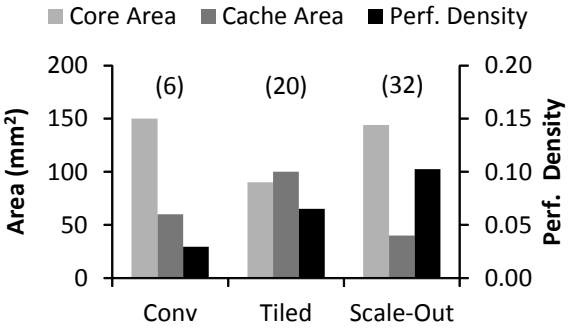


Figure 4. Performance density comparison of various designs. The number above each design corresponds to the number of cores integrated in the design.

workload suite for the three processor organizations. For all designs, the performance is directly proportional to performance density due to the similar area of the evaluated designs. The number above each group of bars indicates the core count of the corresponding design.

We model designs executed in 40nm technology, with a die area of approximately 280mm^2 and impose equal constraints on chip power and bandwidth. To simplify the comparison, we assume out-of-order cores for all three processor types (although the microarchitecture is more aggressive in the conventional design). Additional details on each of the organizations can be found in Section 5.

The conventional chip organization achieves low performance density and therefore, low performance, due to the misallocated core and cache area budget. High core complexity carries a significant area cost, but does not translate into a commensurate performance gain. The LLC is over-provisioned for the low core count and modest capacity demands of the secondary working sets of scale-out applications.

The tiled design features considerably higher execution resources compared to the conventional chip, as lower core area and complexity affords more cores on a die within the imposed area and power limits. As a result, the tiled organization achieves 2.2x higher performance per unit area. However, performance density falls short of optimal because, similar to conventional designs, the LLC is over-provisioned for scale-out workloads, while the multi-hop interconnect introduces latency overheads that diminish performance.

Finally, the Scale-Out configuration, based on the methodology proposed in this work, attains the highest performance density among the evaluated designs. The scale-out design optimizes area utilization by limiting the cache capacity and dedicating more area for cores. The pod-based organization further improves performance through lower LLC access latencies due to shorter interconnect delays.

Table 1. Area and power estimates for various system components at 40nm.

Component		Area	Power
Cores	Conventional	25mm^2	11W
	OoO	4.5mm^2	1W
	In-order	1.3mm^2	0.48W
LLC	16-way SA	5mm^2 per MB	1W per MB

Component	Area	Power
Interconnect	0.2 - 4.5 mm^2	<5W
DDR3 interface (PHY+ controller)	(2 + 10) mm^2	5.7W
SoC components	42mm^2	5W

5. Methodology

We compare the performance, area, and energy efficiency of scale-out processors to conventional and tiled server chips using a combination of cycle-accurate simulation, analytic models, and technology studies.

5.1. Design and Technology Parameters

Baseline (40nm). We compare the various chip architectures in 40nm technology with an on-chip supply voltage of 0.9V. We model chips with an area of 250-280mm^2, power budget of 95W, and a maximum of six single-channel DDR3 interfaces; these parameters are representative of existing server processors fabricated in 40 and 45nm process technology.

Design parameters are summarized in Table 1. We consider three types of cores and a range of cache sizes. Conventional processors feature an aggressive, 4-wide, large-instruction-window core microarchitecture. Tiled and scale-out designs are assessed using two types of cores: (1) high-performance three-way out-of-order core, similar to ARM Cortex-A15 [24], and (2) dual-issue in-order core, resembling ARM Cortex-A8 [1]. To simplify the comparison, we assume a 2GHz operating frequency for all three core types. Cache parameters are estimated using CACTI 6.5 [20].

We estimate the area of the memory interfaces and other SoC components by scaling the micrograph of a Nehalem processor in 45nm technology [17]. We measure the power consumption of a DDR3-1667 channel to be 5.7W. Assuming effective utilization of 70% [4], a 12.8GB/s channel provides 9GB/s of useful bandwidth. We estimate the power of other SoC components and interfaces to be 5W using McPAT v0.8 [18] configured to model Sun UltraSparc T2.

Scaling study (20nm). To understand the effect of technology scaling on the different processor configurations, we also model our systems in 20nm technology. We assume perfect area scaling of cores and caches over two technology generations. Per ITRS estimates, we model a supply voltage of 0.8V and choose not to increase the frequency to minimize power dissipation. We find that the analog circuitry in the PHYs prevents memory interfaces from truly benefitting from technology scaling. We evaluate systems with two types of memory interfaces: existing DDR3 and the emerging DDR4 interface, which is expected to double per-channel memory bandwidth over DDR3 [12].

Table 2. Baseline chip organizations at 40nm.

Processor Design	Cores	LLC (MB)	MC	Die (mm^2)	Power (Watt)	PD
Conventional	6	12	2	276	94	0.030
Tiled (OoO)	20	20	2	257	56	0.065
Tiled (In-order)	64	20	2	251	67	0.112

5.2. Chip Organizations

Table 2 summarizes the parameters for the baseline chip organizations. Conventional chip design is representative of existing products. The Tiled in-order organization is similar to the Tilera Tile64 processor [26]. The number of memory channels in Tiled and Scale-Out chips is computed to accommodate the worst-case bandwidth demand across the workload spectrum for every core/cache organization. For all chip compositions, we model as many cores as can be afforded without exceeding the area, energy, and bandwidth constraints specified in Section 5.1. Performance estimation methodology is described in Section 5.4.

Conventional: 2MB of LLC per core. Cores and caches are interconnected via a crossbar. One DDR3 channel for every four cores. In 40nm technology, six cores can be afforded without exceeding the 95W power budget.

Tiled with OoO cores: 1MB of LLC per tile. The OoO Tiled design is area-limited; 20 cores can be integrated on a 280mm^2 die while maintaining a regular grid topology with a reasonable aspect ratio.

Tiled with in-order cores: The Tiled design with in-order cores maintains the same core-to-cache area ratio of the OoO design. The resulting configuration affords 64 cores and 20MB of LLC, with area as the limiting factor.

All Tiled designs use a mesh interconnect with a per-hop delay of 3 cycles (router + link).

Scale-Out: Cache capacity and core count are determined by evaluating a broad design space from 1 to 256 cores and LLC capacities in the range of 1 to 8MB. Results are presented in Section 6.

Table 3. System parameters for cycle-accurate full-system simulations.

CMP Size	1-64 cores (Data Serving, MapReduce, SAT Solver),1-32 cores (Web Frontend, Web Search)
Processing Cores	*Conventional:* 4-wide dispatch/retirement, 128-entry ROB, 32-entry LSQ, 2GHz *Out-of-order:* 3-wide dispatch/retirement, 60-entry ROB, 16-entry LSQ, 2GHz *In-order:* 2-wide dispatch/retirement, 2GHz
L1I / D Caches	*Conventional:* 64KB, 4(8)-way I(D) cache, 3-cycle load-to-use, 2 ports, 32 MSHRs *Rest:* 32KB, 2-way, 2-cycle load-to-use, 1 port, 32 MSHRs
Last-Level Cache	16-way set-associative, 64B lines, 64 MSHRs, 16-entry victim cache *UCA:* 1 bank per 4 cores; *NUCA:* 1 bank per tile
Interconnect	*Ideal crossbar*: 4 cycles *Crossbar*: 1-8 cores: 4 cycles; 16, 32, 64 cores: 5, 7, and 11 cycles respectively *Mesh*: 3 cycles/hop (includes both router and channel delay)
Main Memory	45ns access latency

5.3. Scale-Out Applications

Our workloads, which include Data Serving, MapReduce, SAT Solver, Web Frontend, and Web Search, are taken from CloudSuite 1.0 [3, 7]. We present two MapReduce workloads: Text Classification and Word Count (referred to as MapReduce-C and MapReduce-W, respectively). For the Web Frontend workload, we use the banking option from SPECweb2009 in place of its open-source counterpart from CloudSuite, as SPECweb2009 exhibits better performance scalability at high core counts. All systems we evaluate run the Solaris 10 operating system.

5.4. Performance Evaluation

We use Flexus [25] for cycle-accurate full-system simulation of various CMP configurations. Flexus extends the Virtutech Simics functional simulator with timing models of in-order and out-of-order cores, caches, on-chip protocol controllers, and interconnect. We model CMPs with 1 to 64 cores, various cache sizes, and three different on-chip interconnects. The details of the simulated architecture are listed in Table 3. Flexus models the SPARC v9 ISA and is able to run unmodified operating systems and applications.

We use the SimFlex multiprocessor sampling methodology [25]. Our samples are drawn over an interval of 10 seconds of simulated time. For each measurement, we launch simulations from checkpoints with warmed caches and branch predictors, and run 100K cycles (2M cycles for Data Serving) to achieve a steady state of detailed cycle-accurate simulation before collecting measurements for the subsequent 50K cycles. We use the ratio of the number of application instructions committed per cycle to the total number of cycles (including the cycles spent executing operating system code) to measure performance; this metric has been shown to accurately reflect overall system throughput [25]. Performance measurements are computed with 95% confidence with an average error of less than 4%.

Because finding optimal pod configurations for scale-out chips requires evaluating a large design space, we augment our simulation-based studies with an analytic model to limit the extent of time-intensive simulation. Our model extends the classical average memory access time analysis to predict per-core performance for a given LLC capacity; the model is parameterized by simulation results, including core performance, cache miss rates, and interconnect delay.

6. Results

We now compare scale-out processor designs to conventional and tiled organizations. For each scale-out design, we first find a performance-density-optimal pod organization, then integrate pods up to the area, energy, and bandwidth limits per Section 5.1. We start by validating our analytic performance model against results obtained via cycle-accurate simulation.

6.1. Model Validation

Figure 5 illustrates the performance density results for designs with out-of-order cores, a 4MB LLC, and different interconnect types across our scale-out applications. We model three different interconnects: a *mesh*, an *ideal crossbar* with a constant delay that is independent of the number of interconnected components, and a realistic crossbar (labeled *crossbar*) whose delay is a function of the core count. The markers in the graph show cycle-accurate simulation results, whereas the lines correspond to the analytic model.

In general, the analytic model predicts performance with excellent accuracy up to 16 cores. At 32 and 64 cores, the actual performance diminishes on three of the workloads (Data Serving, Web Search, and SAT Solver) due to poor software scalability, an effect not captured by the model. Performance scales well with core count for the remaining three workloads, and our model shows good accuracy even

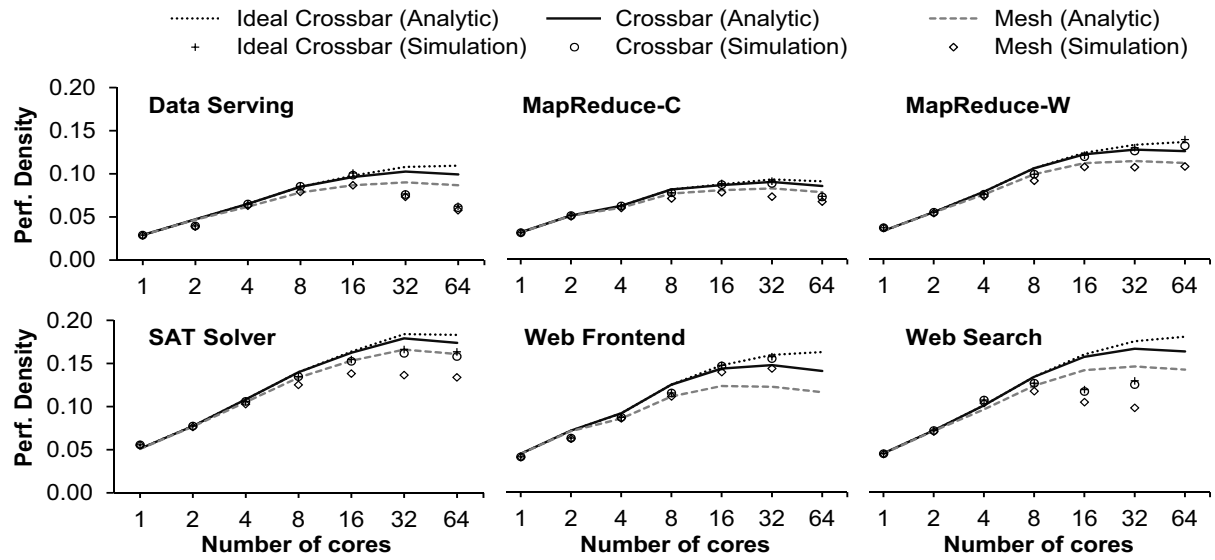


Figure 5. Cycle-accurate simulation and analytic results for designs with out-of-order cores and a 4MB LLC.

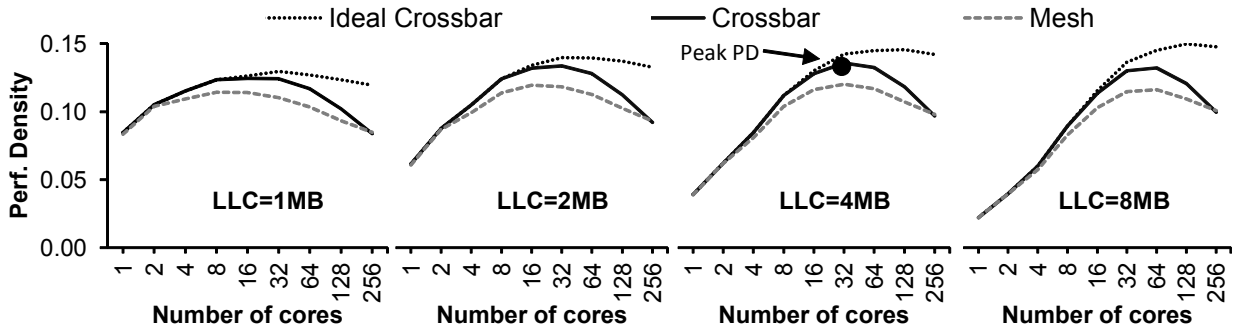


Figure 6. Performance density for a system with out-of-order cores and a range of last-level cache sizes.

at high core counts. The only exception is the Web Frontend application in a mesh-based design, which is less sensitive to interconnect delays than the model predicts.

6.2. System with Out-of-Order Cores

We begin our study with out-of-order cores in the 40nm technology. Figure 6 plots performance density, averaged across all applications, for four different LLC sizes. We do not consider cache sizes above 8MB, as bigger caches do not provide performance improvements (Section 2). Each graph consists of three lines, corresponding to one of the three interconnects described in Section 6.1.

In systems with a realistic interconnect (i.e., crossbar or mesh), performance density starts diminishing above 32 cores regardless of cache capacity, indicating that the physical distance between the cores and the LLC hurts performance when integrating a large number of cores.

Performance density is maximized with 32 cores, a 4MB LLC, and a crossbar interconnect. However, the peak is almost flat. In such cases, software scalability bottlenecks, coherence complexity, and the difficulty of implementing a crossbar interconnect for a large number of cores is likely to shift the design toward a near-to-optimal pod with fewer cores.

To explore this trade-off, Figure 7 examines performance density of pods based on a crossbar interconnect across various LLC sizes. Among designs with fewer than 32 cores, a pod which integrates 16 cores and 4MB of LLC is within 6% of the true optimum. We therefore adopt the 16-core 4MB LLC design with a crossbar interconnect as the preferred pod configuration due to its high PD at modest design complexity.

The PD-optimal pod occupies 92mm^2 and draws 20W of power for cores, caches, and the crossbar. Peak bandwidth demand is 9.4GB/s for 16 cores.

Chip-level assessment. Under constraints specified in Section 5.1, a scale-out processor can afford two pods before hitting the area limit. The resulting chip features 32 cores on a 263mm^2 die with a TDP of 62W.

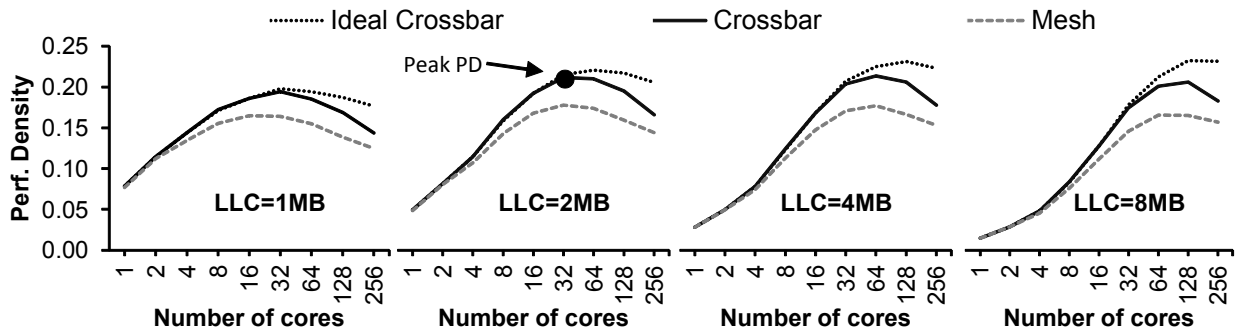


Figure 8. Performance density for a system with in-order cores and a range of last-level cache sizes.

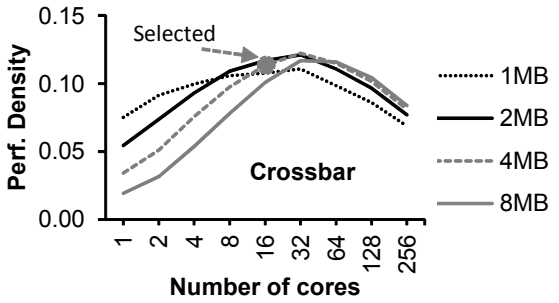


Figure 7. Performance density of pods (OoO) based on a crossbar interconnect and various LLC sizes.

Compared to the conventional design (see Table 2), a scale-out processor with out-of-order cores achieves nearly 3.4x higher performance density, thanks to its greater execution resources, resulting from lower-complexity cores and a smaller LLC. Performance density of the scale-out design is 1.6x higher when compared to the tiled architecture. The latter is hampered by over-provisioned cache capacities and excessive LLC access delays stemming from the multi-hop topology. As Figure 6 shows, a tiled design even with a optimally-sized LLC (i.e., 4MB) shows an 11% lower PD at 32 cores as compared to the crossbar-based dual-pod configuration selected for the scale-out processor.

6.3. System with In-Order Cores

In the recent years, there has emerged a trend towards simpler cores in server processors. Companies like Tilera target scale-out datacenters with chips based on simple in-order cores, validating prior research showing that such designs are well suited for certain scale-out workloads [19]. Although server processors based on simple cores may sacrifice latency, for services whose primary concern is throughput (e.g., data analysis), high-throughput designs that integrate many simple cores may be preferred to organizations with fewer cores of higher performance. Following this trend, we continue our study with in-order cores in 40nm technology.

Figure 8 illustrates performance density results, averaged across all workloads, for cache sizes ranging from 1 to 8MB and three different interconnects. The general trends are similar to those described in the previous section; however, simpler cores in a throughput-oriented architecture yield an optimal pod design with 32 cores, 2MB of LLC, and a crossbar interconnect. To mitigate the complexity associated with a very large crossbar, pairs of cores can share a switch interface. Because the per-core performance is lower than in a design with out-of-order cores, we find the impact of switch sharing to be negligible.

The PD-optimal pod occupies 52mm^2 and draws 17W of power for cores, caches, and the crossbar. Peak bandwidth demand is 15GB/s for 32 cores.

Chip-level assessment. A scale-out processor with in-order cores integrates three PD-optimal pods and is area-constrained. With memory interfaces and miscellaneous peripheral circuitry factored in, this configuration requires 270mm^2 of die area and a TDP of 91W.

The scale-out chip achieves a 1.5x improvement in performance density over a tiled design with in-order cores, and a 5.8x improvement over a conventional processor. Compared to a tiled design with an optimally-sized LLC, a scale-out processor improves performance density by 27%.

6.4. Projection to 20nm Technology

To understand the effect of technology scaling on the scale-out processor design, we project our systems with out-of-order and in-order cores to the 20nm technology node. In the tiled design, per-hop delays remain the same as in the 40nm baseline, as greater wire RC delays are offset by a reduction in tile dimensions. In scale-out chips, the same phenomenon ensures that optimal pod organizations are effectively unchanged under technology scaling, although per-core performance is slightly improved due to a 30% reduction in cache access latency.

In a scale-out processor based on out-of-order cores, 7 pods can be integrated for a total of 112 cores. The resulting configuration is area-limited, occupying 251mm^2, dissi-

pates 83W of power at peak load, and achieves a performance density of 0.378, an improvement of 3.7x over the 40nm baseline. While ideal scaling predicts PD to improve by a factor of 4, the growth in area dedicated to the memory interfaces that do not benefit from technology scaling reduces the fraction of the die available to compute and dampens the gains in PD.

Compared to the scale-out organization, a technology-scaled conventional design is power-limited at 20nm. It integrates 12 cores on a die, improving performance density by a factor of 2.5 from the 40nm design. The tiled architecture with out-of-order cores enjoys ideal scaling in core count, reaching 80 cores in an area-limited configuration, with performance density growing by 3.6x over the 40nm baseline. Performance improvements in the tiled organization is constrained by the growth in the network diameter, which increases LLC access delays. Of the three designs, the scale-out architecture shows the strongest scalability, improving performance density by 5x and 1.6x over conventional and tiled architectures, respectively, when implemented in 20nm technology.

In an implementation based on in-order cores, the scale-out configuration is bandwidth-limited, assuming a constraint of six memory controllers on a die. Compared to the 40nm baseline, the number of pods doubles to six on 192mm^2 die. Technology scaling improves performance density by a factor of 2.8, instead of the expected factor of 4, due to the large area overheads introduced by the on-die memory interfaces necessary to feed the scale-out chip.

Unlike the scale-out design, the tiled organization is power-limited at 20nm. Compared to the 40nm design, the tiled chip has 2.8x more cores (180 vs. 64) and 2.3x higher performance density. In absolute terms, the scale-out processor improves performance density by 6.5x and 1.9x over conventional and tiled architectures respectively, when both are engineered with in-order cores in the 20nm technology.

The analysis above assumes the use of DDR4 memory interfaces. If DDR3 interfaces are used instead, scale-out designs using both core types, as well as tiled chips with in-order cores, will be bandwidth-limited at 20nm. Our results corroborate prior work showing that highly-integrated chips based on simple cores will necessitate bandwidth-boosting techniques, such as 3D-stacked DRAM caches, to mitigate the memory bandwidth wall [10].

6.5. Summary

Table 4 summarizes chip-level features, power and bandwidth requirements, performance/Watt, and performance/mm^2 (i.e., performance density) for the various processor designs. Under an area-normalized comparison, processors with a higher performance density necessarily yield higher performance. Conversely, for a given performance target, PD-optimized designs need a smaller die area compared to chips with a lower performance density.

The conventional architecture achieves the lowest performance density among the evaluated designs at both 40 and 20nm technology nodes. Conventional designs have low performance density because (a) the caches are over-provisioned, allowing less area for compute; and (b) the compute area is misallocated, as aggressive cores provide only a small performance improvement over less aggressive out-of-order cores, yet consume considerably more area.

Tiled organizations using a mesh-based interconnect and out-of-order cores achieve 2.2x higher performance density than conventional in 40nm technology (3.1x in 20nm). The use of lower-complexity cores improves performance density, and as a result, throughput; however, the large LLC and the delays associated with a multi-hop interconnect limit the performance gains.

The highest performance density is achieved in a scale-out processor, which uses a pod-based organization to limit interconnect delays and maximizes compute area through modestly-sized last-level caches. A scale-out design with out-of-order cores improves performance density by 3.4x and 1.6x over conventional and tiled chips, respectively, in 40nm technology (5x and 1.6x over the respective designs in 20nm).

On workloads with laxer QoS requirements, higher performance density (and consequently, higher throughput) can be achieved through the use of in-order cores. In such cases, a scale-out chip improves performance density by 5.8x (6.5x) and 1.5x (1.9x) over conventional and tiled designs, respectively, in 40nm (20nm) technology. Our results corroborate prior work, which shows that low-complexity cores are well-suited for throughput applications [5, 11]. The results also underscore scale-out processors' advantage under technology scaling, as both in-order and out-of-order scale-out configurations improve the lead in performance density over conventional and tiled chips as technology is scaled from 40 to 20nm.

Finally, we note that scale-out organizations are effective in improving processor energy-efficiency, in addition to performance density. Compared to tiled organizations, performance per watt is improved by 1.5x, and 1.2x for out-of-order and in-order designs, respectively, in 40nm technology. The improvements extend to 1.5x and 1.7x at 20nm. Energy efficiency in scale-out chips is improved through higher per-core performance and lower energy/op. While core efficiency is the same for scale-out and tiled chips with the same core type, scale-out chips dissipate less energy in the memory hierarchy through smaller caches (less leakage) and smaller communication distances.

Table 4. Performance density, area, power, and bandwidth requirements of various processor designs.

	40nm							20nm						
Processor design	PD	Cores	LLC (MB)	MCs	Die (mm^2)	Power (Watt)	Perf/ Watt	PD	Cores	LLC (MB)	MCs	Die (mm^2)	Power (Watt)	Perf/ Watt
Conventional	0.030	6	12	2	276	94	0.09	0.075	12	48	3	213	93	0.17
Tiled (OoO)	0.065	20	20	2	257	56	0.30	0.233	80	80	2	256	80	0.75
Scale-Out (OoO)	0.103	32	8	3	263	62	0.44	0.378	112	28	4	251	83	1.14
Tiled (In-order)	0.112	64	20	2	251	67	0.42	0.258	180	80	4	249	94	0.68
Scale-Out (In-order)	0.173	96	6	6	270	91	0.51	0.491	192	12	6	192	80	1.18

7. Related Work

The notion that scale-out workloads benefit from a many-core architecture was advocated by Hardavellas et al. [10], who argued for use of simpler cores and minimum on-die caches provided there is no bandwidth bottleneck. Our work extends that idea by introducing a scalable and efficient implementation of such a many-core architecture.

In order to reduce access time to the LLC, researchers have proposed Non-Uniform Cache Architectures (NUCA) [15]. One way to overcome the interconnect delays in the resulting organization is through richly-connected topologies [16]; however these have been shown to have significant area and energy overheads in many-core chips [8]. In this work, we show that a pod-based design with a simple crossbar interconnect and a modestly-sized LLC overcomes the inefficiency of NUCA designs on scale-out workloads.

Prior work that tried to find the optimal CMP design either focused on finding the optimal cache architecture for a given core count [13, 28], or on finding the optimal core microarchitecture for a given application [6]. Most of the prior work assumes non-datacenter applications [6, 22]. Oh et al. [22] presented a simple and effective analytic model to study the core count versus cache capacity trade-off in CMPs under die area constraints, showing that for a fixed cache size, increase in core count hurts the aggregate performance beyond a certain point. Our work corroborates this result on scale-out applications.

Prior research and industry efforts have attempted to maximize the compute area by reducing the fraction of the die allocated to cache. Kgil et al. [14] proposed eliminating last-level caches and devoting their area to cores, while compensating for the increase in memory bandwidth pressure through 3D-stacked DRAM. Similarly, we find that increasing the fraction of the chip dedicated to compute is important for throughput; however, we also observed that scale-out workloads have reuse in their secondary working set and benefit from a modest cache size. Graphics processors (GPUs) also use a large number of processing elements with minimal cache resources. For instance, Tesla C1060 GPUs have 240 processing elements with under 756KB of aggregate cache capacity [21]. GPU architectures are tuned for high throughput and are unlikely to satisfy latency demands of real-time online services.

Certain commercial and research chips share some of the insights or conclusions of this work. Piranha [2] was the first chip multi-processor designed for commercial server workloads that used simple cores for higher efficiency. In this work, we also showed that using simple cores for scale-out workloads is beneficial from a performance density perspective. Sun Niagara III is a contemporary server processor that, at a high level, resembles a scale-out pod in that it features 16 cores and a 6MB LLC connected via a crossbar switch [23]. However, the cores are 8-way multithreaded, resulting in poor single-threaded performance and high area overhead. In addition, Niagara chips have not adopted a multi-pod design, instead scaling-up capabilities through additional resources.

8. Conclusions

Server processors for scale-out datacenters must maximize silicon efficiency by carefully balancing core, cache and interconnect considerations. While large caches degrade performance through long access delays, modestly-sized last-level caches enable fast instruction fetches and maximize the area available for the cores.

Existing server chip architectures make sub-optimal use of the die real-estate. Conventional server chips use inappropriately-complex core microarchitectures, given scale-out workload characteristics. Tiled many-core chips feature more efficient cores, but lose performance in the multi-hop interconnect. Both architectures provide excessively large caches, diminishing silicon area available for compute.

To overcome the limitations of existing server chips, we introduced a processor design methodology based on pods, performance-density-optimal processor building blocks. Each pod couples a small last-level cache to a number of cores using a low-latency interconnect. Pod-based designs maximize per-processor throughput in today's technology and provide a simple mechanism to scale the designs to future technologies without diminishing performance den-

sity. Using a combination of model-driven analysis and cycle-accurate simulation, we demonstrate that scale-out processors with out-of-order cores achieve 5x and 1.6x higher performance density over conventional and tiled processors, respectively.

9. Acknowledgments

We thank the EuroCloud project partners for inspiring the scale-out processors. This work was partially supported by EuroCloud, Project No 247779 of the European Commission 7th RTD Framework Programme – Information and Communication Technologies: Computing Systems.

References

[1] M. Baron. The F1: TI's 65nm Cortex-A8. *Microprocessor Report*, 20(7):1–9, July 2006.

[2] L. A. Barroso, K. Gharachorloo, R. McNamara, A. Nowatzyk, S. Qadeer, B. Sano, S. Smith, R. Stets, and B. Verghese. Piranha: a scalable architecture based on single-chip multiprocessing. In *Proc. of the Int'l Symposium on Computer Architecture*, June 2000.

[3] CloudSuite 1.0. http://parsa.epfl.ch/cloudsuite.

[4] H. David, C. Fallin, E. Gorbatov, U. R. Hanebutte, and O. Mutlu. Memory power management via dynamic voltage/frequency scaling. In *Proc. of the ACM Int'l Conference on Autonomic Computing*, June 2011.

[5] J. D. Davis, J. Laudon, and K. Olukotun. Maximizing CMP throughput with mediocre cores. In *Proc. of the Int'l Conference on Parallel Architectures and Compilation Techniques*, Sept. 2005.

[6] M. Ekman and P. Stenstrom. Performance and power impact of issue-width in chip-multiprocessor cores. In *Proc. of the Int'l Conference on Parallel Processing*, Oct. 2003.

[7] M. Ferdman, A. Adileh, O. Kocberber, S. Volos, M. Alisafaee, D. Jevdjic, C. Kaynak, A. D. Popescu, A. Ailamaki, , and B. Falsafi. Clearing the clouds: a study of emerging scale-out workloads on modern hardware. In *Proc. of the Int'l Conference on Architectural Support for Programming Languages and Operating Systems*, Mar. 2012.

[8] B. Grot, J. Hestness, S. W. Keckler, and O. Mutlu. Kilo-NOC: a heterogeneous network-on-chip architecture for scalability and service guarantees. In *Proc. of the Int'l Symposium on Computer Architecture*, June 2011.

[9] N. Hardavellas, M. Ferdman, B. Falsafi, and A. Ailamaki. Reactive NUCA: near-optimal block placement and replication in distributed caches. In *Proc. of the Int'l Symposium on Computer Architecture*, June 2009.

[10] N. Hardavellas, M. Ferdman, B. Falsafi, and A. Ailamaki. Toward dark silicon in servers. *IEEE Micro*, 31(4):6–15, Jul-Aug 2011.

[11] N. Hardavellas, I. Pandis, R. Johnson, N. Mancheril, A. Ailamaki, and B. Falsafi. Database servers on chip multiprocessors: limitations and opportunities. In *The Conference on Innovative Data Systems Research*, Jan. 2007.

[12] JEDEC Announces Key Attributes of Upcoming DDR4 Standard. http://www.jedec.org/news/pressreleases/jedec-announces-key-attributes-upcoming-ddr4-standard. 2011.

[13] N. P. Jouppi and S. J. E. Wilton. Tradeoffs in two-level on-chip caching. In *Proc. of the Int'l Symposium on Computer Architecture*, Apr. 1994.

[14] T. Kgil, S. D'Souza, A. Saidi, N. Binkert, R. Dreslinski, T. Mudge, S. Reinhardt, and K. Flautner. PicoServer: using 3D stacking technology to enable a compact energy efficient chip multiprocessor. In *Proc. of the Int'l Conference on Architectural Support for Programming Languages and Operating Systems*, Oct. 2006.

[15] C. Kim, D. Burger, and S. W. Keckler. An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches. In *Proc. of the Int'l Conference on Architectural Support for Programming Languages and Operating Systems*, Oct. 2002.

[16] J. Kim, W. J. Dally, and D. Abts. Flattened butterfly: a cost-efficient topology for high-radix networks. In *Proc. of the Int'l Symposium on Computer Architecture*, June 2007.

[17] R. Kumar and G. Hinton. A family of 45nm IA processors. In *IEEE Int'l Solid-State Circuits Conference*, Feb. 2009.

[18] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi. McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proc. of the Int'l Symposium on Microarchitecture*, Dec. 2009.

[19] K. Lim, P. Ranganathan, J. Chang, C. Patel, T. Mudge, and S. Reinhardt. Understanding and designing new server architectures for emerging warehouse-computing environments. In *Proc. of the Int'l Symposium on Computer Architecture*, June 2008.

[20] N. Muralimanohar, R. Balasubramonian, and N. Jouppi. Optimizing NUCA organizations and wiring alternatives for large caches with CACTI 6.0. In *Proc. of the Int'l Symposium on Microarchitecture*, Dec. 2007.

[21] NVIDIA Tesla Computing Processor. http://www.nvidia.com/docs/IO/43395/NV_DS_Tesla_C1060_US_Jan10_lores_r1.pdf.

[22] T. Oh, H. Lee, K. Lee, and S. Cho. An analytical model to study optimal area breakdown between cores and caches in a chip multiprocessor. In *Proc. of the IEEE Computer Society Annual Symposium on VLSI*, May 2009.

[23] J. L. Shin, K. Tam, D. Huang, B. Petrick, H. Pham, C. Hwang, H. Li, A. Smith, T. Johnson, F. Schumacher, D. Greenhill, A. S. Leon, and A. Stron. A 40nm 16-core 128-thread CMT SPARC SoC processor. In *IEEE Int'l Solid-State Circuits Conference*, Feb. 2010.

[24] J. Turley. Cortex-A15 "Eagle" flies the coop. *Microprocessor Report*, 24(11):1–11, Nov. 2010.

[25] T. F. Wenisch, R. E. Wunderlich, M. Ferdman, A. Ailamaki, B. Falsafi, and J. C. Hoe. SimFlex: statistical sampling of computer system simulation. *IEEE Micro*, 26(4):18–31, Jul-Aug 2006.

[26] B. Wheeler. Tilera sees opening in clouds. *Microprocessor Report*, 25(7):13–16, July 2011.

[27] D. A. Wood and M. D. Hill. Cost-effective parallel computing. *IEEE Computer*, 28(2):69–72, Feb. 1995.

[28] L. Zhao, R. Iyer, S. Makineni, J. Moses, R. Illikkal, and D. Newell. Performance, area and bandwidth implications on large-scale CMP cache design. In *Proc. of the Workshop on Chip Multiprocessor Memory Systems and Interconnects*, Feb. 2007.